AGILE PROJECT MANAGEMENT AND DEVELOPMENT
Author: Jim Skinner, PMP, North Carolina Department of Insurance

I first became interested in Agile Software Development last year after reading in a trade journal how industry was beginning to embrace it. As I looked into it, I felt I could apply various Agile methodologies to improve our current development processes and increase scope control, timeliness and quality of future projects. Agile is a team driven iterative approach to applications development in which the users and product owner work closely with the development team throughout the application development life cycle. I became interested in similar methodologies, Joint Application Development (JAD) and Rapid Application Development (RAD) in the 90's while working with a PMO in Sacramento CA but never actually merged the two together as Agile does. JAD and RAD were used in a gated "Waterfall" process like we have now at different stages of the project lifecycle. I used JAD to develop the Software Requirements Specification (SRS) and later used RAD during the build phase, using prototyping to build until a satisfactory model was acceptable. In Agile, these two processes are woven together in an iterative process.

In December 2007, DOI completed the first gate of a new project and registered it in the Project Portfolio Management Tracking System. Our project was the update and conversion of a system that is used to regulate and manage the manufactured housing industry within North Carolina. The current system was a series of Microsoft Access Databases with Access and Visual Basic Applications that interfaced with Excel Spreadsheets used by in-house staff and field inspectors. The new system would be browser based, developed in Java with an Oracle database, combining the functionality of the spreadsheets and the applications. We had already begun the formal requirements gathering. Our usual methodology was that the team would divide the current functions and meet with the subject matter experts to gather the requirements. The team member who met with the subject matter expert became the most knowledgeable of that function when we moved into the build phase. This was a process that I had felt needed improvement. Around mid February, I received a training flyer offering a class in Agile Project Management. The class was a good introduction into the various classifications (Scrum, XP, etc). Upon completing the course, I decided that I wanted to move my development and project management effort at the DOI to Agile. Unfortunately, I had neither funding nor time to bring my team of seven analyst programmers and one database administrator up to speed on Agile. A coach is really recommended to get a team trained and going on an Agile project. When I presented the concepts of Agile to the team, most were very receptive. Some took the "they needed more information" approach. After sharing with them what I brought back from my class, I put it on the back burner until we got a little closer to the end of requirements gathering.

While the team was working with the subject matter experts to finalize the Software Requirements Specification, I was able to get funding for another class for the Scrum Master Certification Workshop. This seemed like the best way to go since funding for the team was not available. I decided I would gather as much as I could and bring it back to the team and provide training sessions as best I could from a three day class.

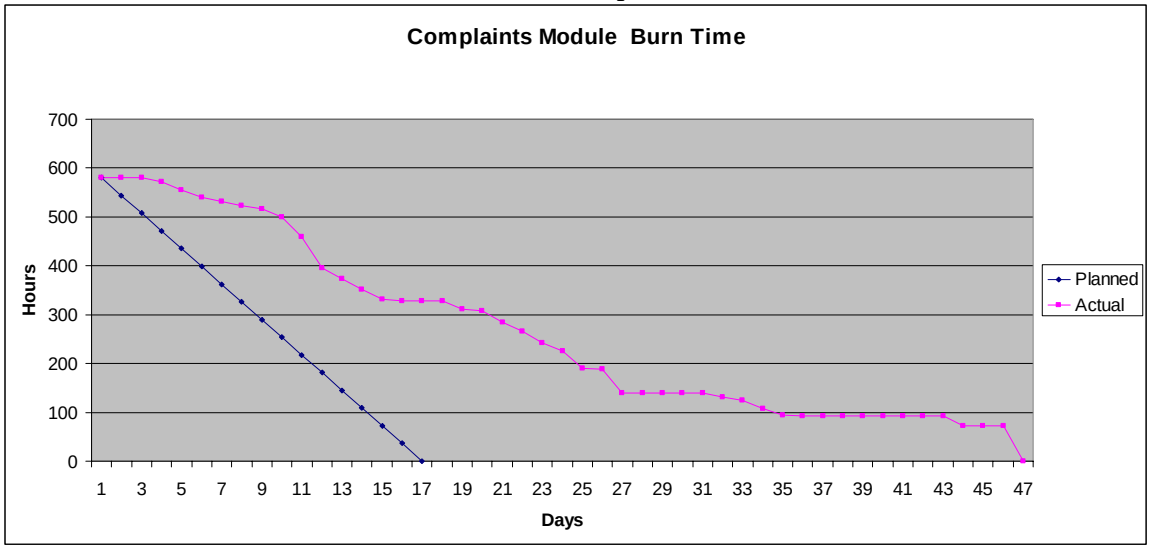
When I sat down with my project owners for sign-off on the SRS, I presented my case for moving to the Agile Methodology. Yes, we were well into the project but it seemed like a good place to move to Agile if we were going to do it on this project. I explained that I would need heavy commitment and involvement from them and their users. I laid it all out. They were very excited to be so involved. Once I knew I had full commitment from the project owner and users, I sat down with the team to let them know that we were in for quite a paradigm shift. In the past each person owned a piece of a project. We would now do it the Agile – Scrum way with every one working together on one area at a time until it was complete and accepted before we moved on. Within Scrum, there is no team leader. The team is self directed and self-managing, self-organizing and cross functional. The team members are collectively responsible for the success of each iteration and to bring it to a satisfactory completion. The Scrum Master is similar to a project manager in that he or she is responsible for removing impediments to the team and making sure Scrum fits within the organizations culture, delivering expected benefits and ensuring everyone follows Scrum rules and practices.

Agile is significantly different from waterfall. It has new terminology to get used to. One of the things I really like is that the Product Owner (the customer) identifies the project requirements in priority order. That is the way the project is approached and the system is built. If for some reason, the project does not run to 100% completion but say 80%, if you have 80% of the project done and the important parts are completed up front, chances are you have a product you can use. Maybe not what was originally envisioned but better than a shelved system with nothing to use. The list of prioritized items is captured in the Product Backlog. The Product Backlog is a high-level requirements document. Since we had completed the formal requirements specification for this project, all we needed was the order of priority. However, going forward in future projects we will do the requirements gathering as Agile recommends. We will develop the Product Backlog so we can prepare our time and cost estimates and better define the requirements as we move through each iteration of development. The Product Backlog will be developed with all team members, the product owner, and the key subject matter experts to capture the high level requirements.

A Sprint Backlog defines the tasks that the team derives from the Product Backlog. "Sprint" is the term for the block of time estimated to complete and deliver a functional portion of the project that satisfies one of the priorities. Since we were very new at this, for our first Sprint, two team members who had completed the requirements analysis in this one area developed the Sprint backlog of tasks and estimated hours to complete per task to determine the length of the Sprint. See cut from a Sprint Backlog below.

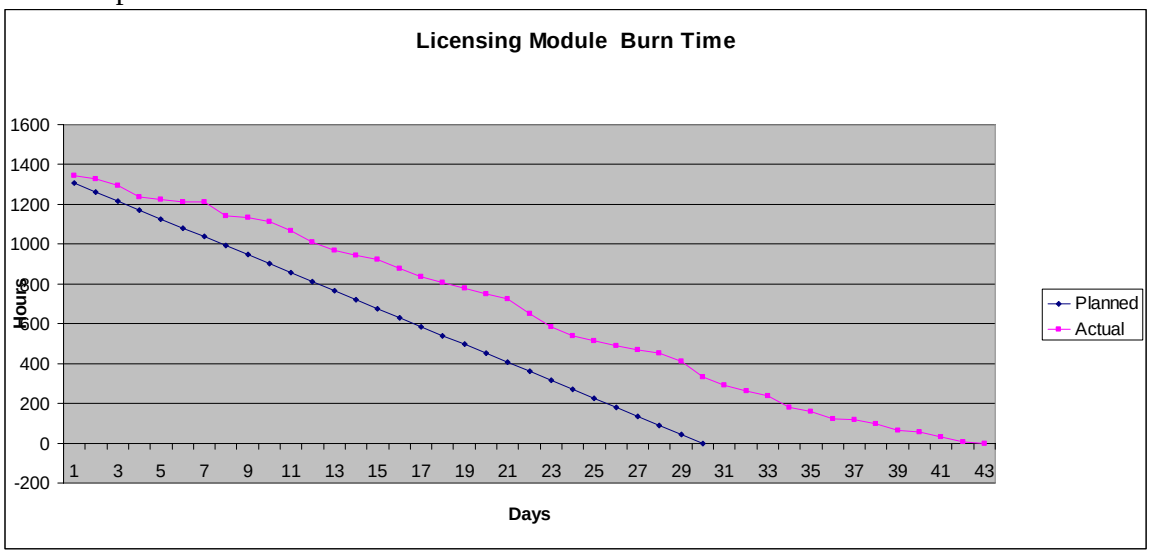
Total Tas				Sprint Da	24	s
Tasks Complete	4					
Percent Complete	10%					
					1	2
					12/15	12/16
TASK LISTINGS	ASSIGNED	STATUS (NS, IP, C)	Planned>>	824	787	749
			Actual		787	743
Monitoring				Planned		
Migration						
Tables creation	Vicente	IP		30	30	27
Data Migration	Vicente	NS		80	80	80
Manufactured Homes (HUD)						
Monitoring						
Dealer Lot Inspection Record	Larry	IP		16	8	4
Dealer Lot entry of dealer Lot questionnaire						
Questions / Exit Interview / Comments Tabs combined	Larry	IP		24	24	24
Print Blank and Completed	Larry	NS		16	16	16
Staff Tab	Torrii	NS		32	32	32
Escrow Tab	Gayle	NS		24	24	24
Homes Inspected Tab	Kiran	IP		24	24	24
Print blank and completed Home Inspection forms	Kiran	NS		16	16	16
Dealer Ref						
Add Homes Inspected Tab to Dealer Ref	Kiran	NS		16	16	16

This action became one of our “Lessons Learned”. The two team members were very knowledgeable of the requirements and tasks for the Sprint. The other six team members needed time to come up to speed. We also had not built in time for the development of the infrastructure which should have been a separate Sprint. This misstep can be seen from viewing the Burn Rate Chart. The “Burn Rate Chart” tracks progress in a Scrum Sprint. The Straight Line is the computed ideal completion rate on a daily basis from beginning to end based on estimations. Above the line is behind schedule and below is ahead of schedule. Note the chart for our first Sprint below.

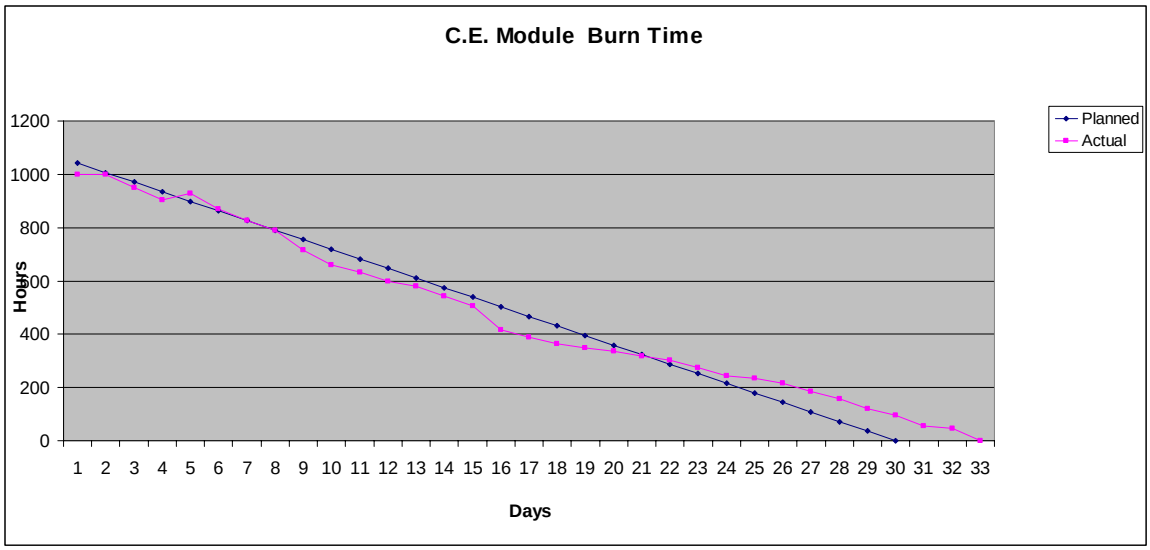


For the second Sprint, again we used two staff members who had developed the requirements for that area of the project. We used the whole team to develop the Sprint Backlog Task list. The work went much better but was still off from the ideal we had estimated. Again, another lesson learned. We thought that since we had involved the users in the requirements gathering, we would not need them to clarify anything before we began. We learned as we worked through the Sprint and met with the users at least once a week or more that we missed a lot up front. The missed requirements changed the

estimates and caused the Sprint to exceed the estimated time. Note the chart from our second Sprint below.



On the third Sprint, we felt we were finally getting the feel for it. The team was getting used to the morning status meetings that are typically short. Each member gives a very brief description of what they have done since yesterdays meeting and what they plan to do before tomorrows meeting. The Burn Down Chart is updated and the meeting is adjourned. Any other issues are discussed as the work is being carried on. So having gone through two Sprints we now knew more involvement with our key users and subject matter experts upfront was needed just as if we had never completed the formal Software Requirements Specification. We spent almost two days with the users going over the requirements to get an in-depth understanding for the third Sprint. The next two days the team worked on updating the Product Backlog for things missed during requirements gathering and developed the sprint backlog. Once the actual development began the project was going great. Each day at the meeting tasks were being completed ahead of schedule. We were below the Burn Line. That was until day 22 of the 30 day Sprint. We had one team member whose spouse was hospitalized for several days unexpectedly. The team member was out for several days. We also had another team member that had planned vacation that had been accounted for in the estimation but it did put us down two resources. Risk management is the same in Agile. As it was, we finished three days over but were getting much better and each piece was finished tested and accepted. Note the Chart from our third Sprint below.



We are currently on our fourth Sprint. We had full involvement with the user community. So far we are slightly ahead of schedule.

Unlike standard project tracking, Agile-Scrum is not Gantt chart-based. The minimum plan necessary to start a Scrum project is a vision and a Product Backlog, though I would recommend a "little more". Our current project was originally estimated to be completed and in production the end of April 2009. By building incrementally by priority and having deliverable working parts that are tested and accepted as we go, we will avoid much of the usual rush to complete, test and accept just before go-live. We are tentatively planning to role out 80% of the prioritized pieces in early February and complete the other 20% (several nice-to-have enhancements and some seldom used functions) later.

For Standards and Testing, we follow the same standards for security and development we have always used. I have modified our testing process based on an Agile white paper I read on the internet. It recommended a buddy coding system where one team member confers with another for a few hours while they work on a task together and review the work as they go. We are still evaluating that one. We also have randomized buddy review for functionality, coding standards and GUI standards. This is just a randomized selection of team members to look at each others work. We also pull pieces for a team review during the week. Finally our Users are involved constantly with incremental testing. We do incremental code updates daily and receive feedback from users via email, face to face meetings or telephone conversations.

My recommendations for moving to Agile would be to try it on a small to medium project. Agile is best suited for small teams, no more than 7-10 people. If working a large project, it is recommended that you break down to small groups and manage a Scrum of Scrums. More can be found on that in the many books available. I would recommend if possible pick a small to medium project to start. Have support from upper management. Have buy-in from the Project owner. Get the team indoctrinated in Agile before starting if at all possible. It puts everyone on the same page from the beginning.

Agile prescribes many processes. All may not work for your group. For instance, Agile calls for the group to work in one room together. I opted not to do that. My group works in the same general area and confers with each other constantly. My experiences have been good and bad with close in. I'm only 65% through my first Agile attempt. We are just beginning the last of the high priority items. I am very pleased with the results and the combined effort of the team. When we begin our next conversion in mid 2009, I feel we will see significant gains.

There are many books beginning to show on the market addressing Agile and Scrum. Two I prefer are AGILE & ITERATIVE DEVELOPMENT A Manager's Guide, by Craig Larman and AGILE PROJECT MANAGEMENT WITH SCRUM by Ken Schwaber. There are many sites with Pro and Con white papers available also on the internet. I truly believe it is the way of the future and is finding its way into structured project management. I have noted over the past few months that there have been several articles and references to Agile in the PM Network Monthly.

For additional information contact:

Jim Skinner, PMP, CSM
N. C. Department Of Insurance
Application Development Manager
(919) 733-5411 ext 303
jskinner@mail.doi.state.nc.us